CISTER - Research Center in
Real-Time & Embedded Computing Systems

Energy Efficient Mapping of Mixed Criticality Applications on Unrelated Heterogeneous Multicore Platforms

Muhammad Ali Awan, Damien Masson*, Eduardo Tovar

CISTER, ISEP, Polytechnic Institutue of Porto, Portugal
*Universite Paris-Est, Paris, France

Outline

- Motivation and challenges
- System model
- Preliminaries
- Proposed allocation heuristics
- Evaluation
- Conclusions and future directions

Motivation and challenges (1)

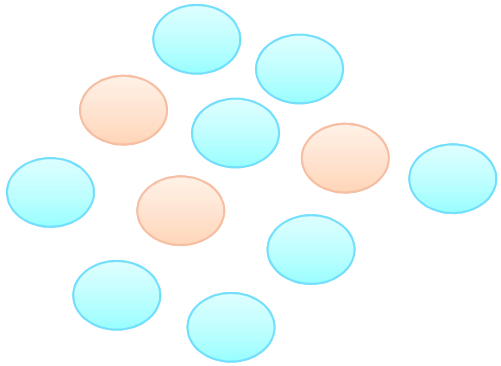
- Modern real-time applications are becoming increasingly complex
  - More than on criticality levels
  - Level of assurance against failure
    - High criticality applications are vital
    - Faults in low criticality tasks are tolerable
- Applications with different criticality levels were hosted in different components
- Trend towards deploying applications with different criticality levels on a same platform
  - Pessimistic WCET estimates
    - Inefficient resource utilisation

Motivation and challenges (2)

- Multicore platforms
  - Extensive computing resources
- Heterogeneous multicore
  - Diverse computing capabilities
  - Perform specific functions efficiently
  - Mapping of mixed criticality applications on a heterogeneous multicore platform is a non-trivial exercise
- Energy is important
  - Size
  - Weight
  - Thermal issue
  - Packing
  - Cost

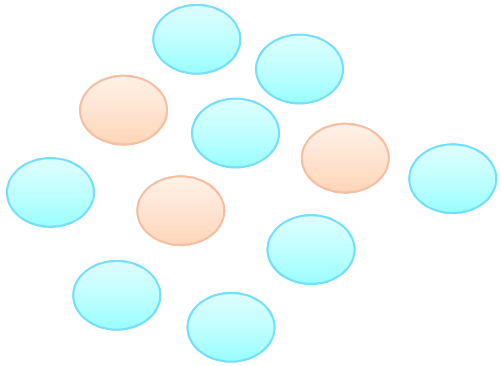
This Work

Mixed criticality applications

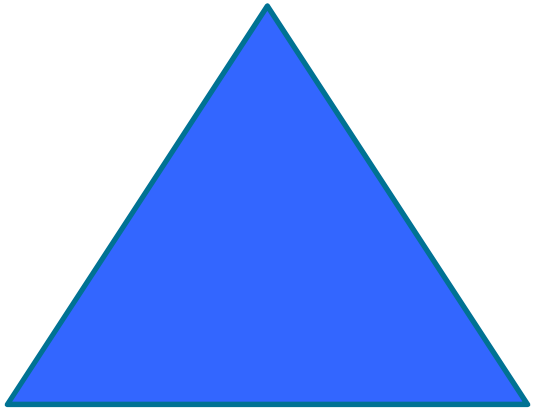


This Work

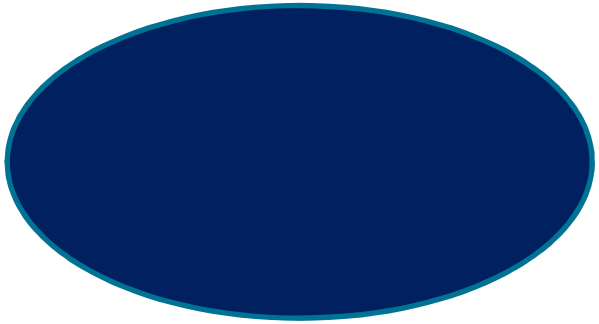
Mixed criticality applications



π^1



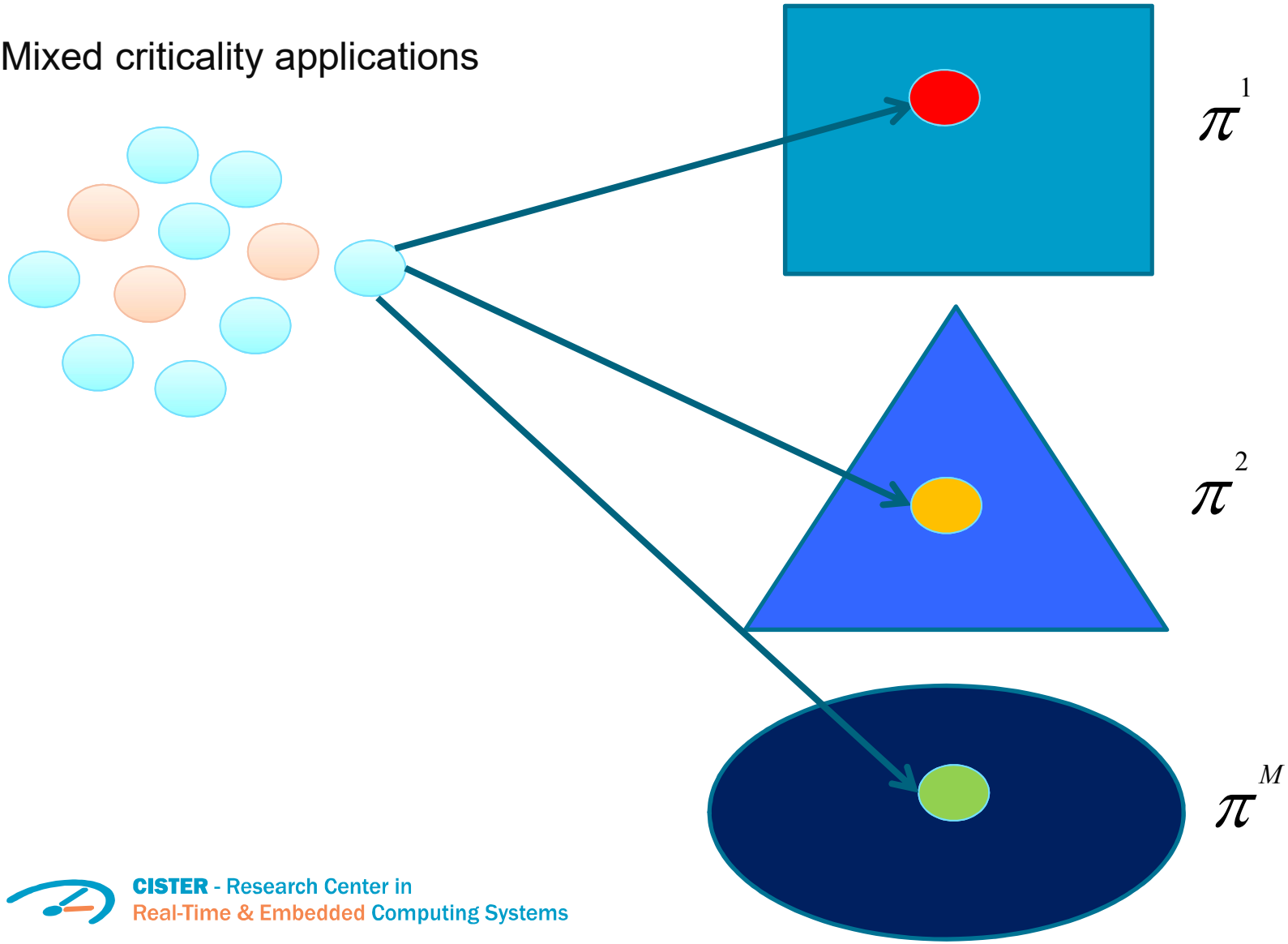
π^2



π^M

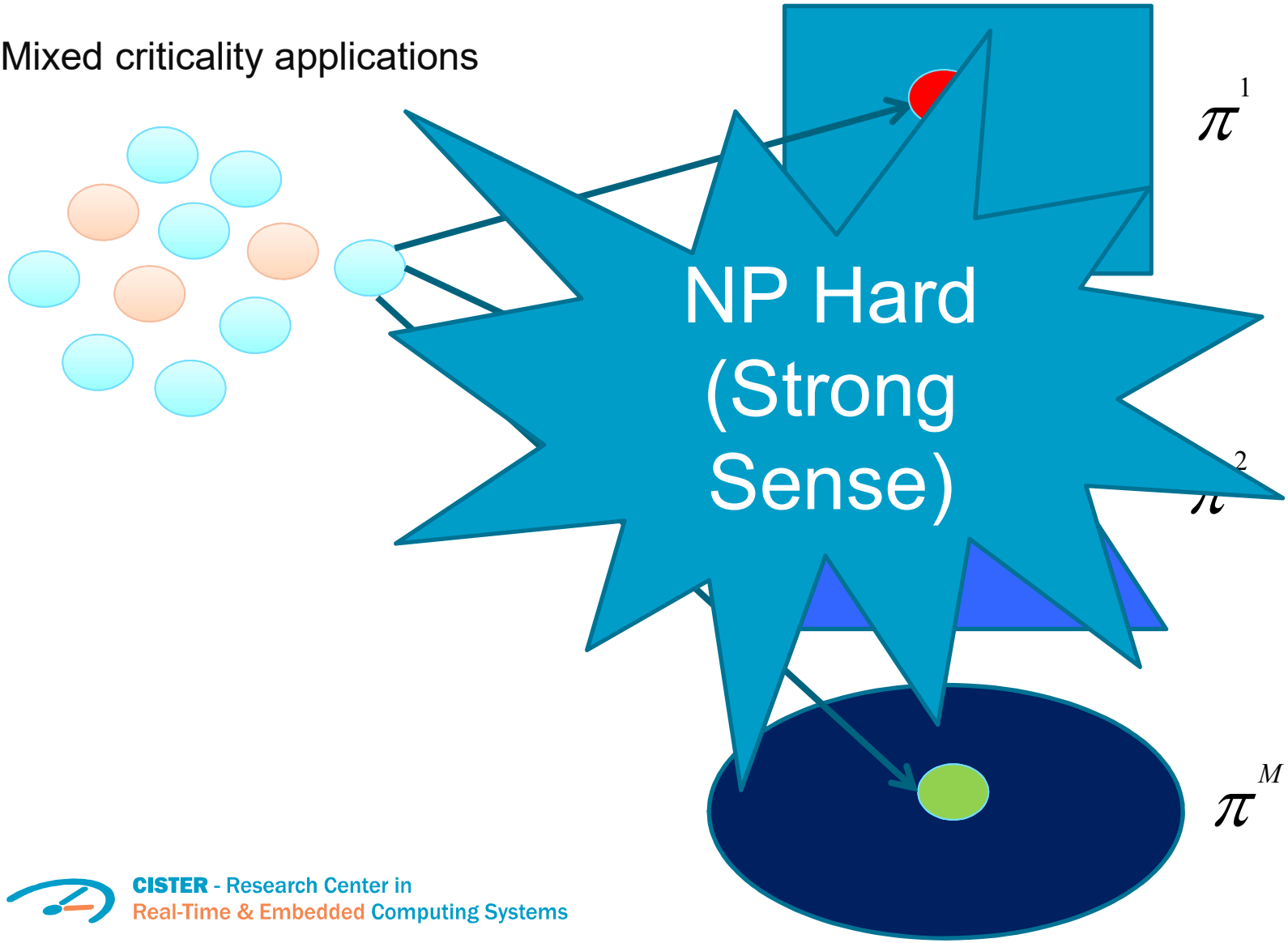
This Work

Mixed criticality applications



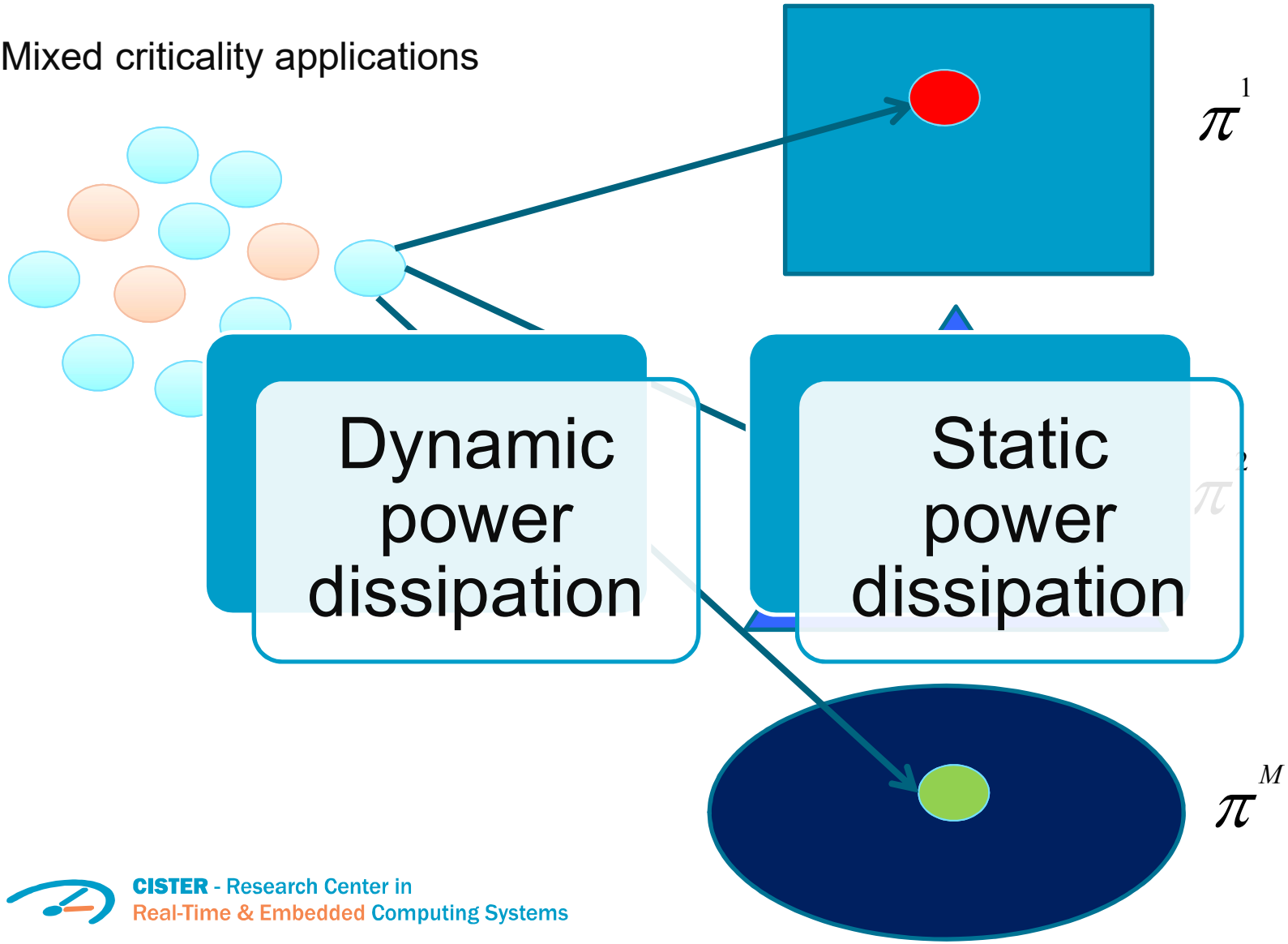
This Work

Mixed criticality applications



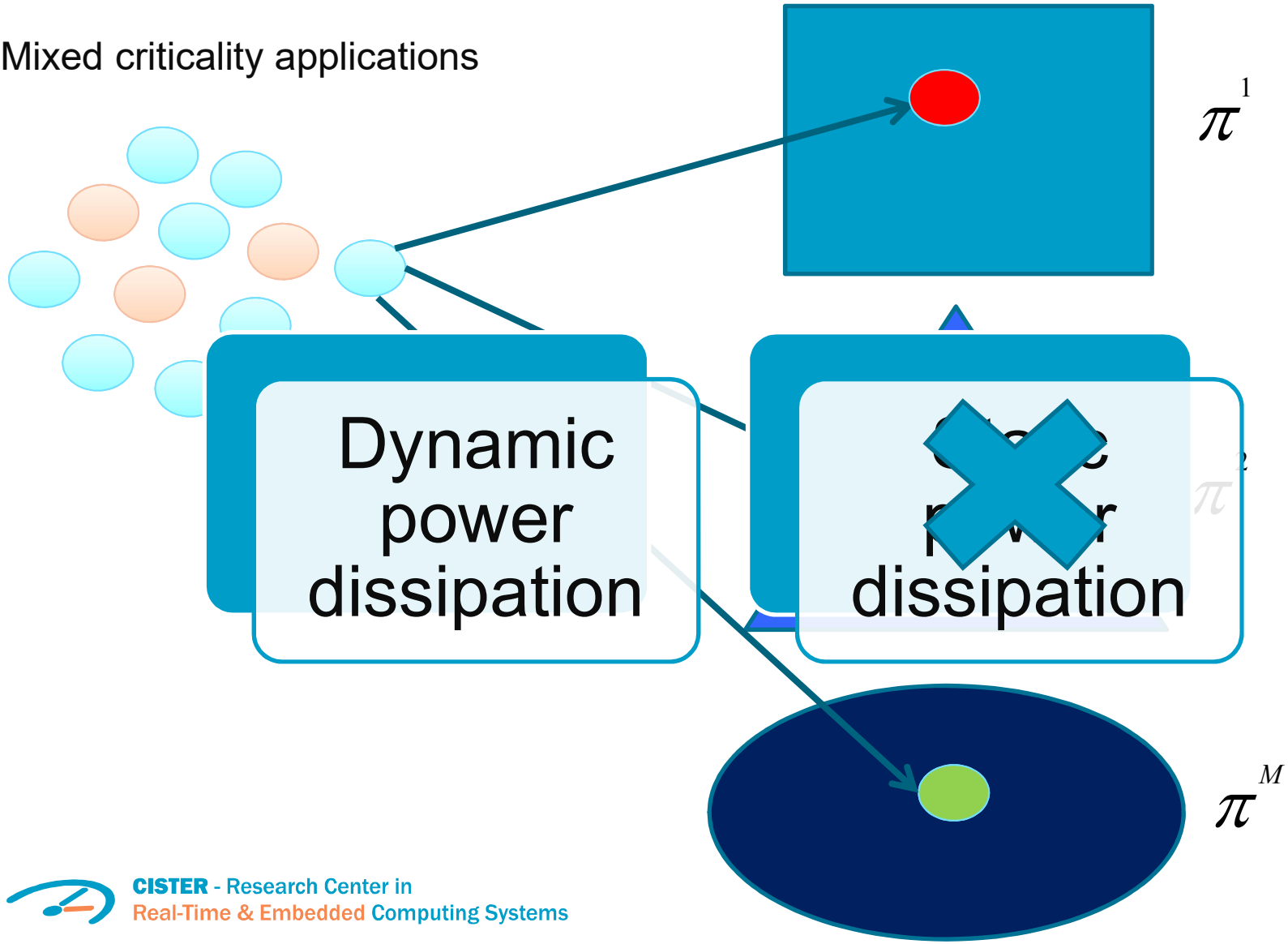
This Work

Mixed criticality applications



This Work

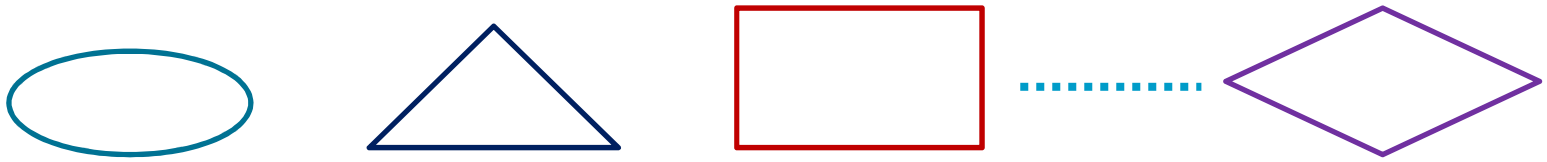
Mixed criticality applications



System model

- Unrelated heterogeneous multicore platform
  - Each core has different characteristics
    - Power dissipation
    - Execution capabilities
  - No relation
- Partitioned scheduling

	Task 1	Task 2
Core 1	2	10
Core 2	5	1



π^1 π^2 π^3 π^M

Vestal's model

- Proposed by Steve Vestal
- Dual criticality
  - Two mode of operation
    - L-mode and H-mode
- System starts in L-mode and transition to H-mode in case of any abnormal behavior
  - Violations of parameters in L-mode
    - Overrun
    - Memory accesses etc

Task model

- Independent tasks
- Each task has either high or low criticality

Task model

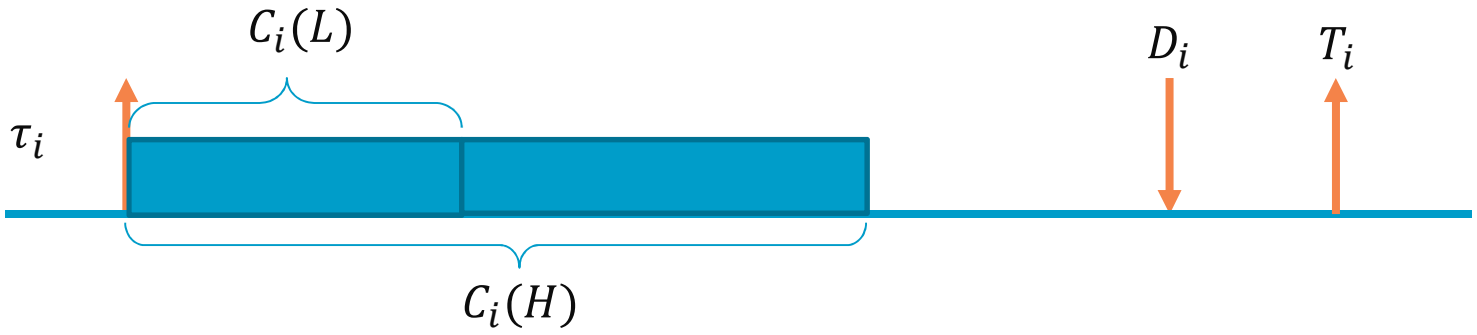
- Independent tasks
- Each task has either high or low criticality

Task model

- Independent tasks
- Each task has either high or low criticality

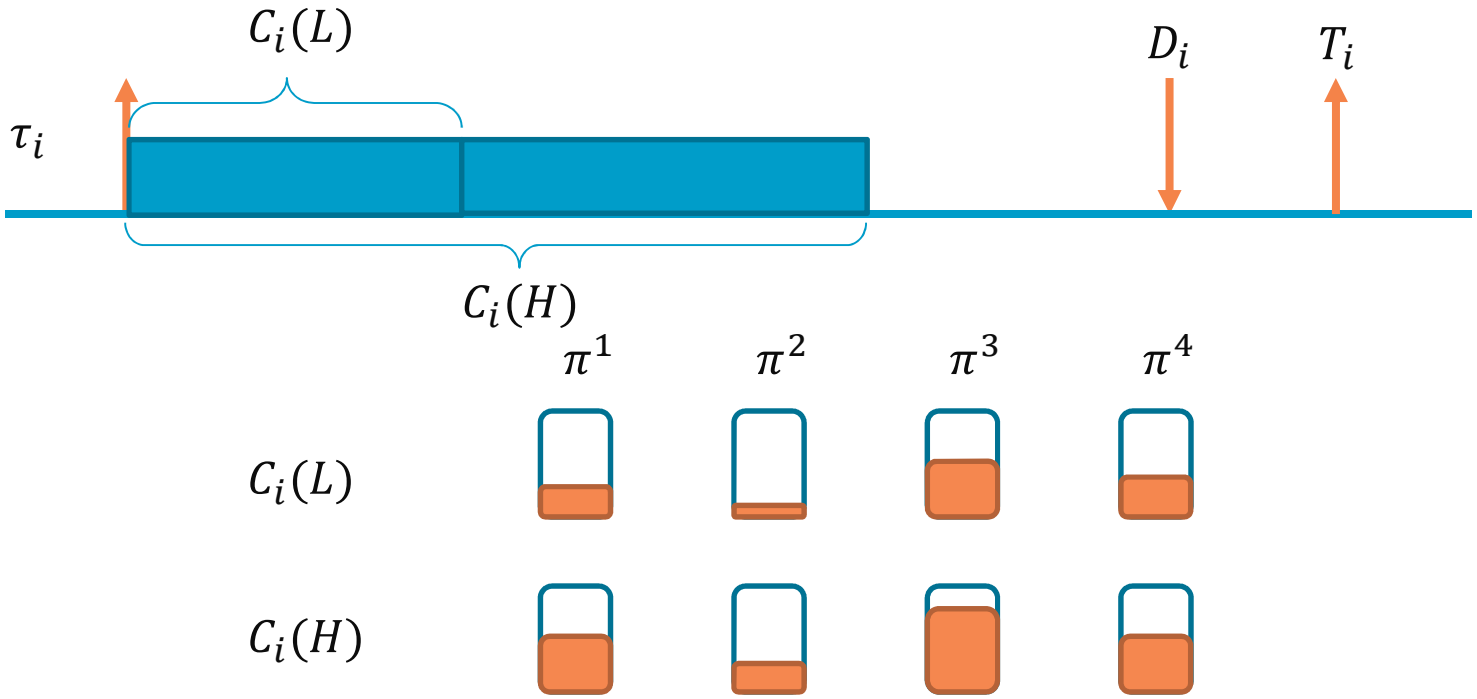
Task model

- Independent tasks
- Each task has either high or low criticality



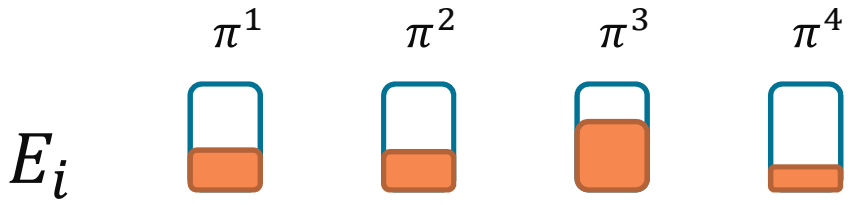
Task model

- Independent tasks
- Each task has either high or low criticality



Energy model

- Energy is not only a function of WCET
  - Depends on the characteristic of the core
    - Set of instruction / parts of core used
  - Two application with similar WCET may have different energy consumption
- Measurement based approach (Snowdon)
  - Incorporates effect of different system resources (cache and memory etc)
- Energy in L-mode of operation



Schedulability analysis

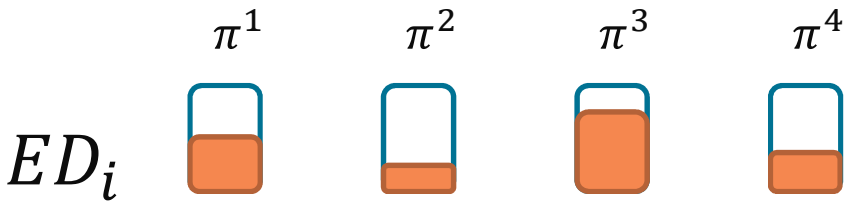
- Mixed criticality schedulability analysis proposed by Ekberg and Yi
  - Ensures schedulability both in low and high mode of operation
  - Demand bound function based analysis
  - Shortens deadline for L-mode of operation
  - Independent scaling of deadlines
  - Ensures schedulability in transition phase
  - Valid for constrained deadline model

Preliminaries (1)

Energy density (ED_i) = $\frac{E_i}{T_i}$

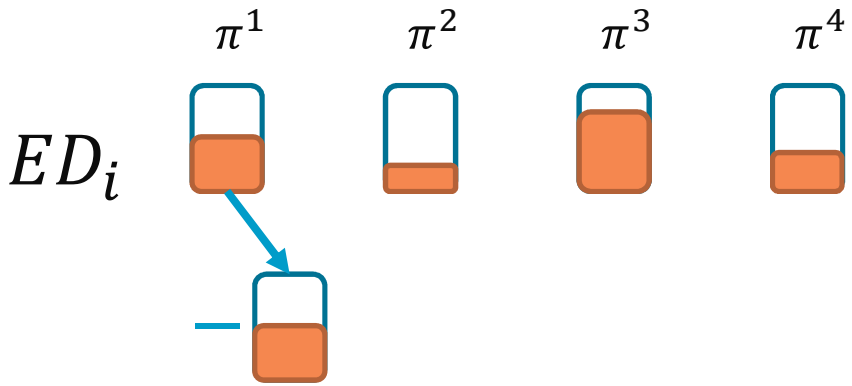
Preliminaries (1)

Energy density (ED_i) = $\frac{E_i}{T_i}$



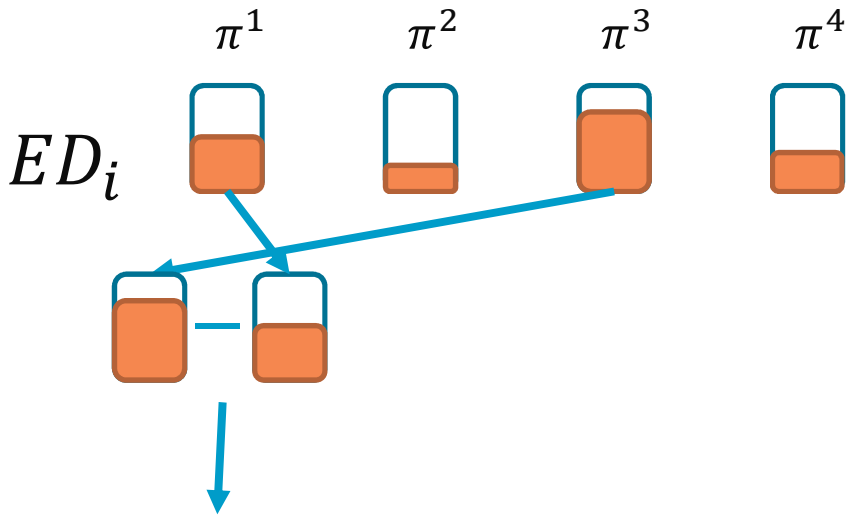
Preliminaries (1)

Energy density (ED_i) = $\frac{E_i}{T_i}$



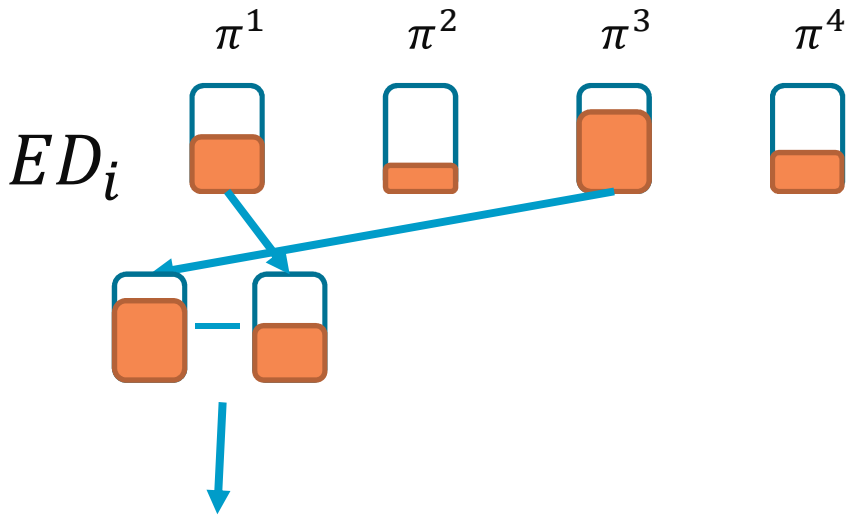
Preliminaries (1)

Energy density (ED_i) = $\frac{E_i}{T_i}$



Preliminaries (1)

Energy density (ED_i) = $\frac{E_i}{T_i}$



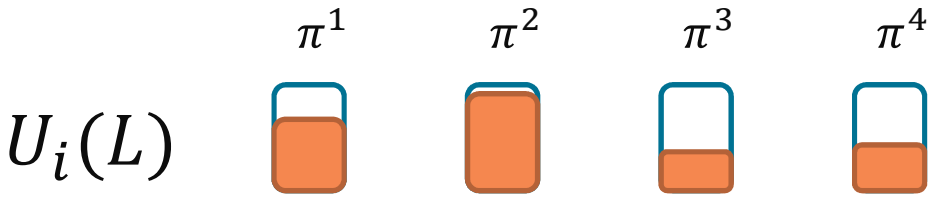
Energy Density Difference EDD_i^1 EDD_i^2 EDD_i^3 EDD_i^4

Preliminaries (2)

Utilisation in L-mode $U_i(L)$ = $\frac{C_i(L)}{T_i}$

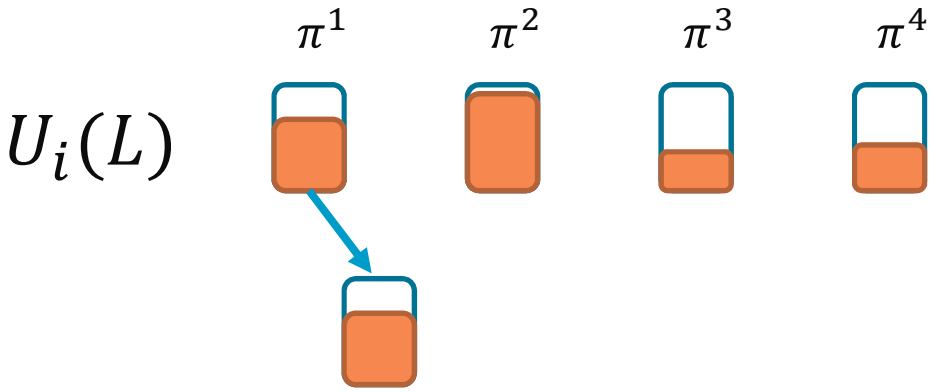
Preliminaries (2)

Utilisation in L-mode $U_i(L)$ = $\frac{C_i(L)}{T_i}$



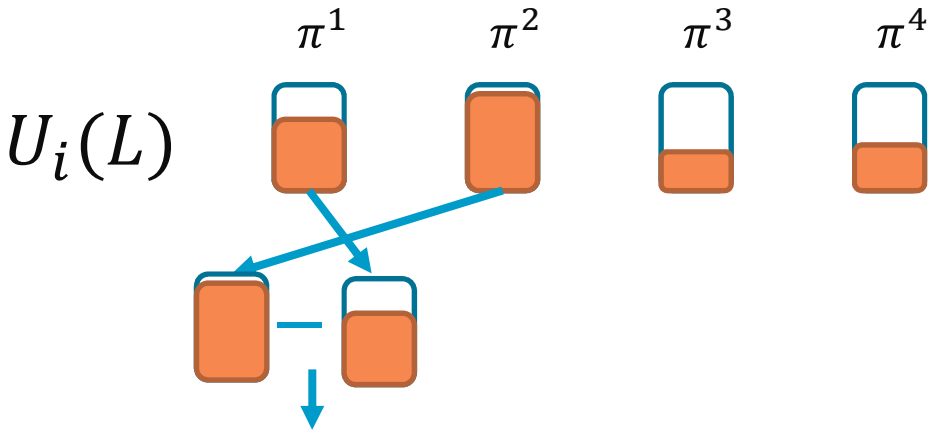
Preliminaries (2)

Utilisation in L-mode $U_i(L)$ = $\frac{C_i(L)}{T_i}$



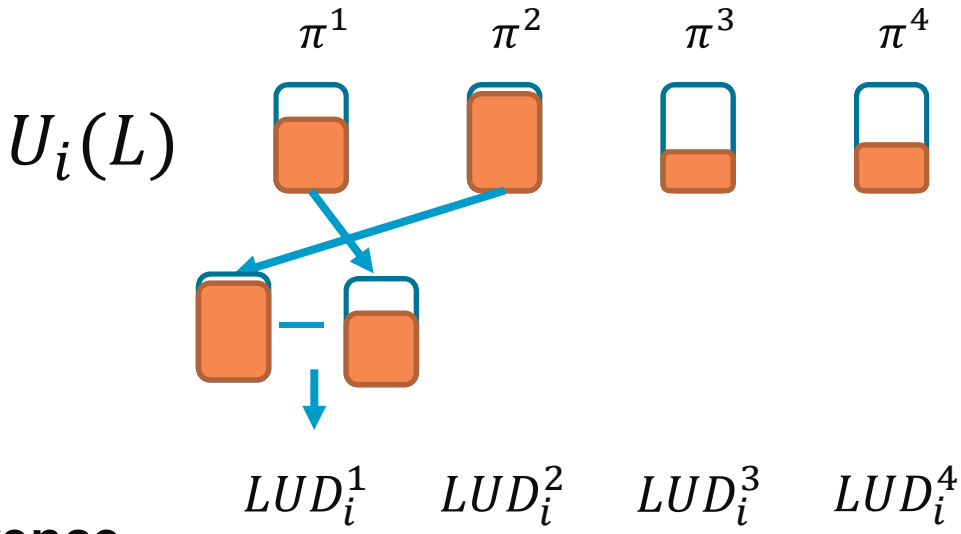
Preliminaries (2)

Utilisation in L-mode $U_i(L)$ = $\frac{C_i(L)}{T_i}$



Preliminaries (2)

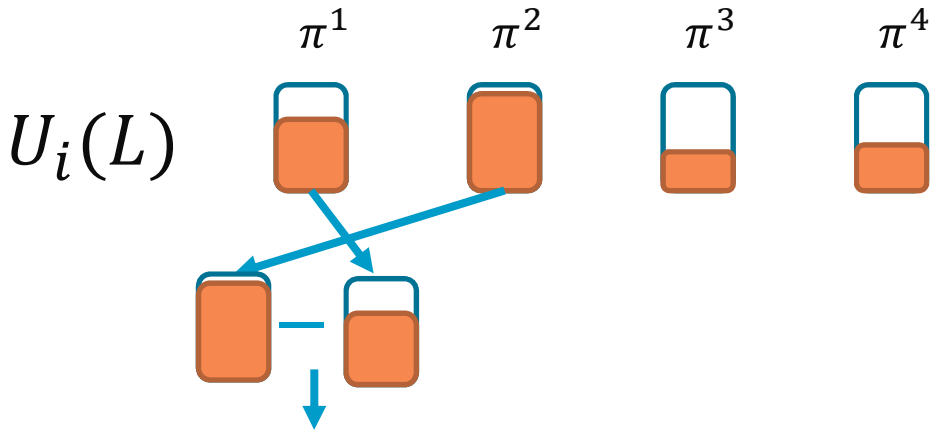
Utilisation in L-mode $U_i(L) = \frac{C_i(L)}{T_i}$



Low criticality utilisation difference

Preliminaries (2)

Utilisation in L-mode $U_i(L) = \dfrac{C_i(L)}{T_i}$



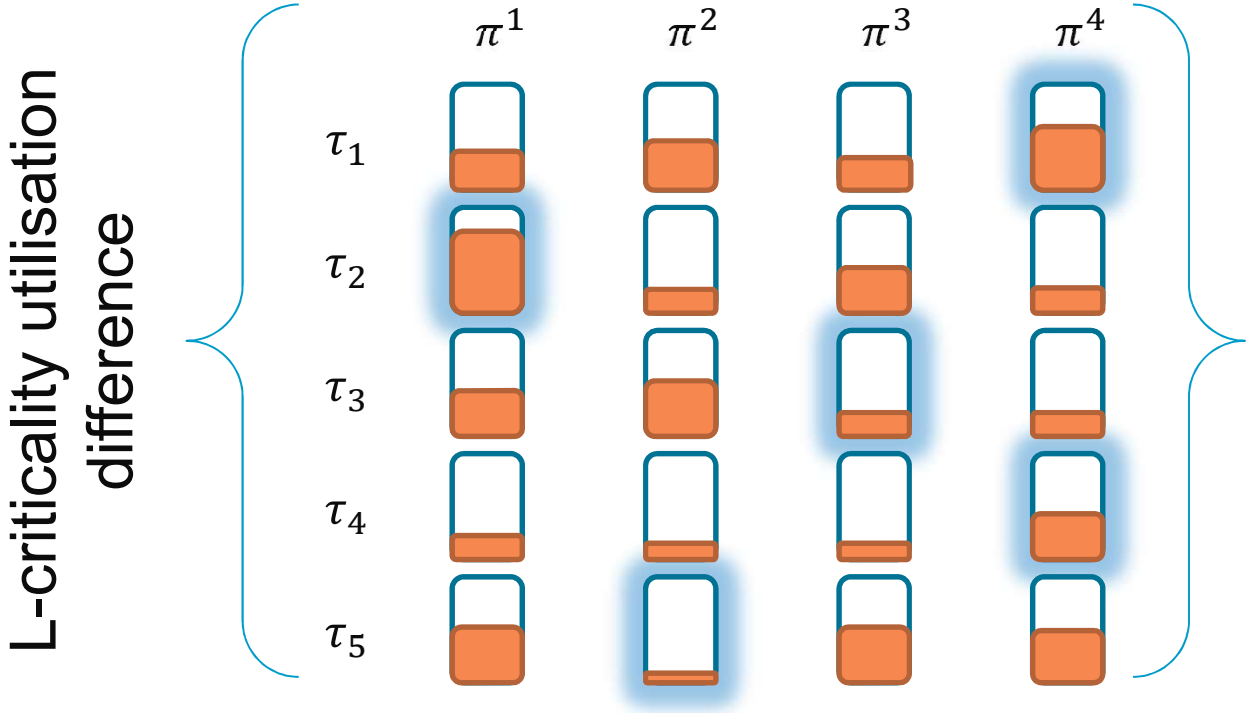
Low criticality utilisation difference LUD_i^1 LUD_i^2 LUD_i^3 LUD_i^4

High criticality utilisation difference HUD_i^1 HUD_i^2 HUD_i^3 HUD_i^4

Preliminaries (3)

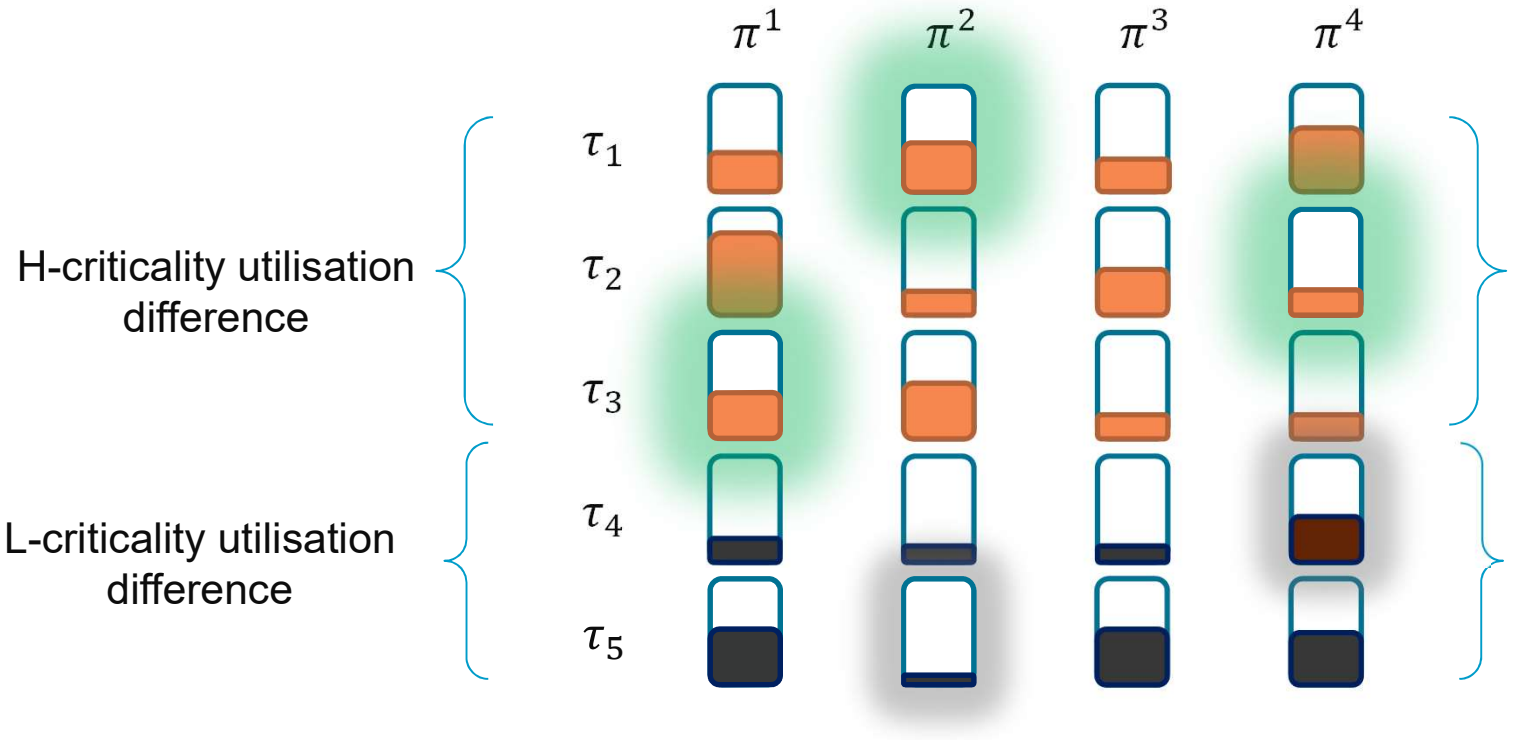
- Favourite core
  - Best option w.r.t any parameter
- Least preferred core
  - Worst option w.r.t any parameter
- Parameters
    - Energy consumption
    - L-mode utilisation
    - H-mode utilisation

Density difference List (1)



- S^{LUD}  Sorted in no-increasing order

Density difference List (2)

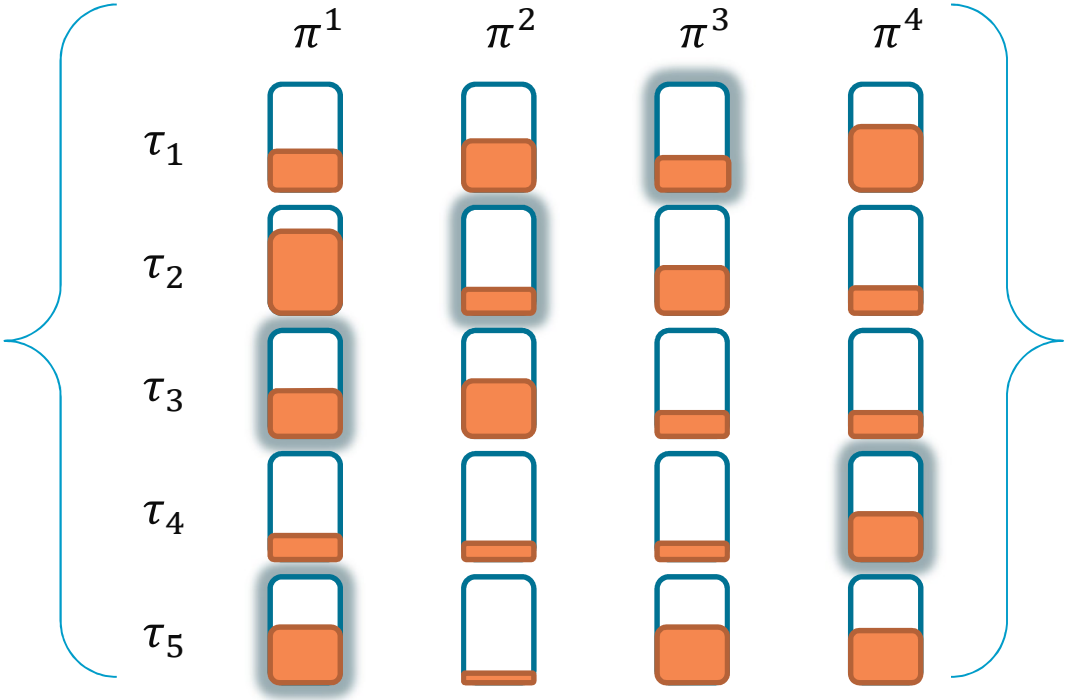


- S^{HUD}

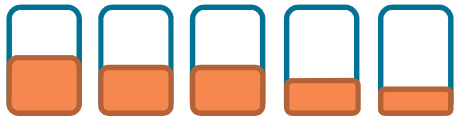
- Sorted list (non-increasing)
  - Criticality first
  - Density difference

Density difference List (3)

Energy density difference



- S^{EDD}



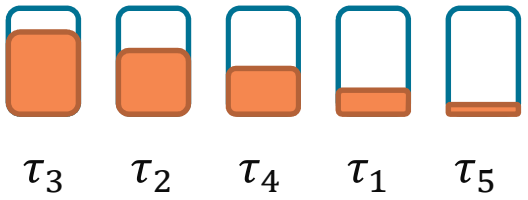
Sorted in no-increasing order of energy density difference

Improved least loss energy density algorithm (ILLED)

- Input: Any density difference list
  - (S^{ED} or S^{LUD} or S^{HUD})
  - Sorted in non-increasing order

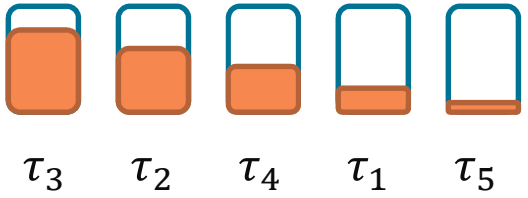
Improved least loss energy density algorithm (ILLED)

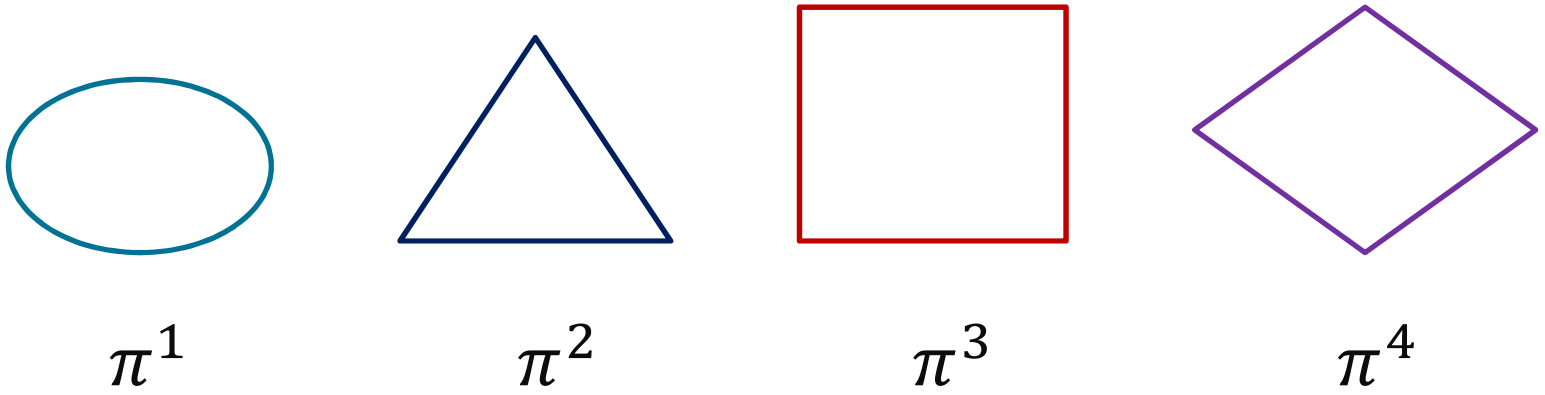
- Input: Any density difference list
  - (S^{EDD} or S^{LUD} or S^{HUD})
  - Sorted in non-increasing order



Improved least loss energy density algorithm (ILLED)

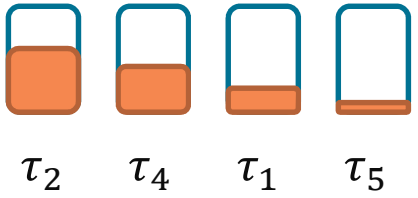
- Input: Any density difference list
  - (S^{EDD} or S^{LUD} or S^{HUD})
  - Sorted in non-increasing order

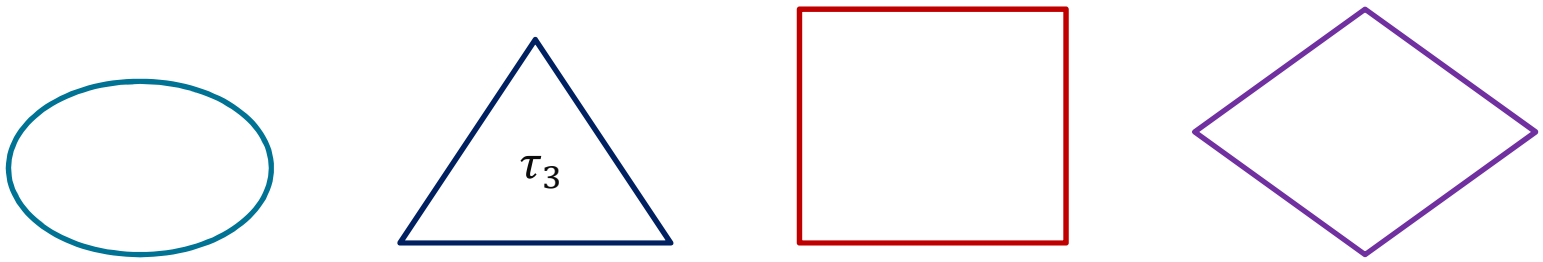




Improved least loss energy density algorithm (ILLED)

- Input: Any density difference list
  - (S^{EDD} or S^{LU} or S^{HUD})
  - Sorted in non-increasing order

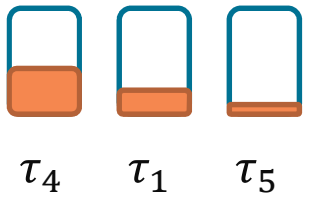


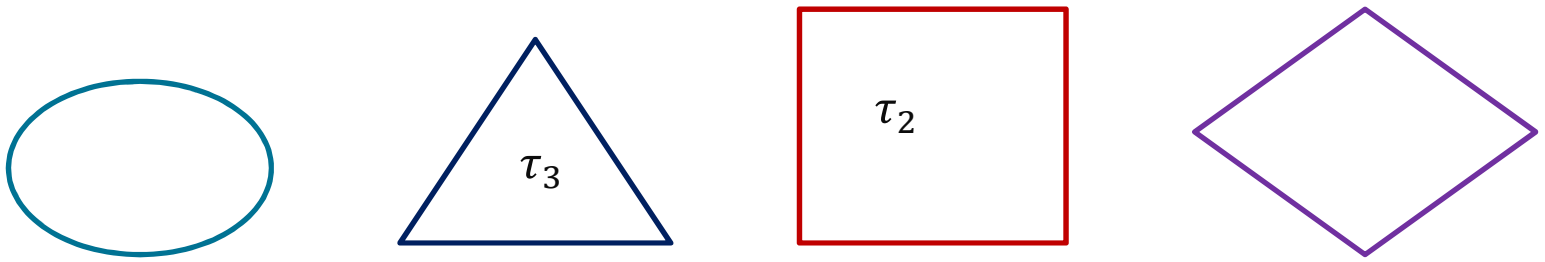


π^1 π^2 π^3 π^4

Improved least loss energy density algorithm (ILLED)

- Input: Any density difference list
  - (S^{EDD} or S^{LUD} or S^{HUD})
  - Sorted in non-increasing order

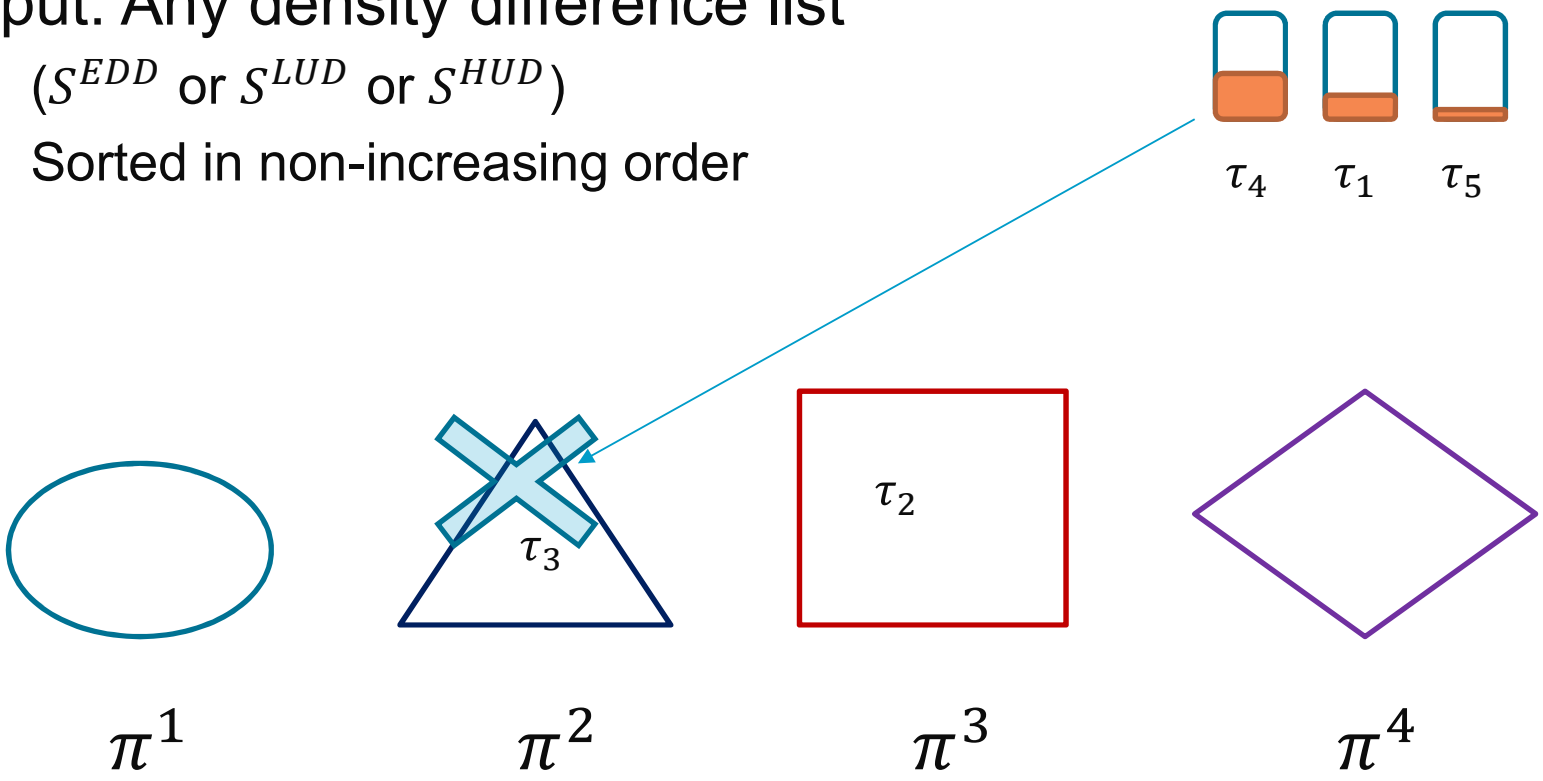




π^1 π^2 π^3 π^4

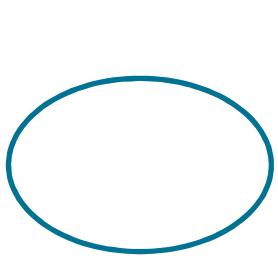
Improved least loss energy density algorithm (ILLED)

- Input: Any density difference list
  - (S^{EDD} or S^{LUD} or S^{HUD})
  - Sorted in non-increasing order

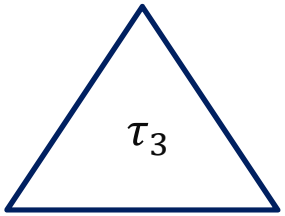


Improved least loss energy density algorithm (ILLED)

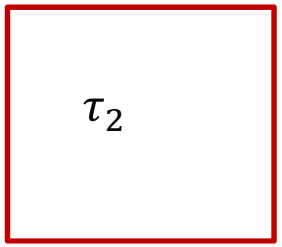
- Input: Any density difference list
  - (S^{ED} or S^{LUD} or S^{HUD})
  - Sorted in non-increasing order



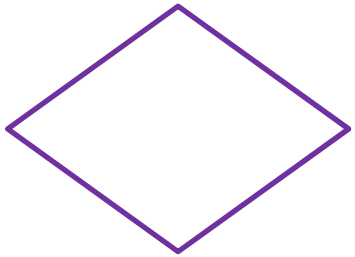
π^1



π^2



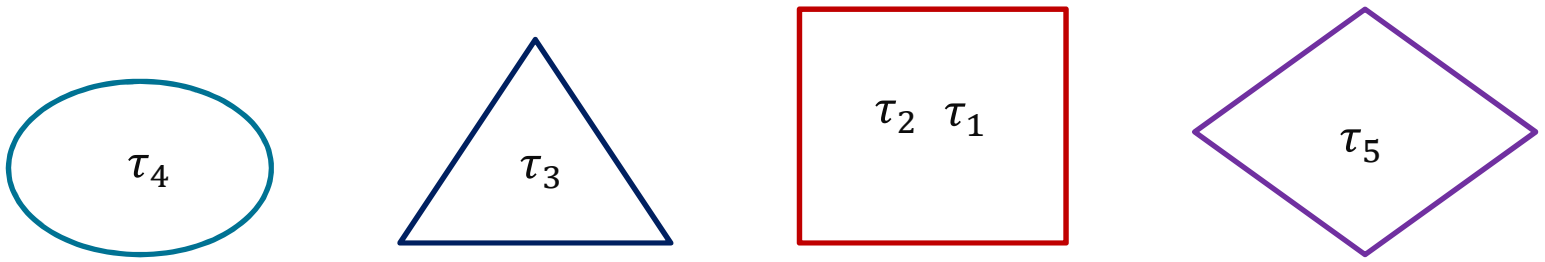
π^3



π^4

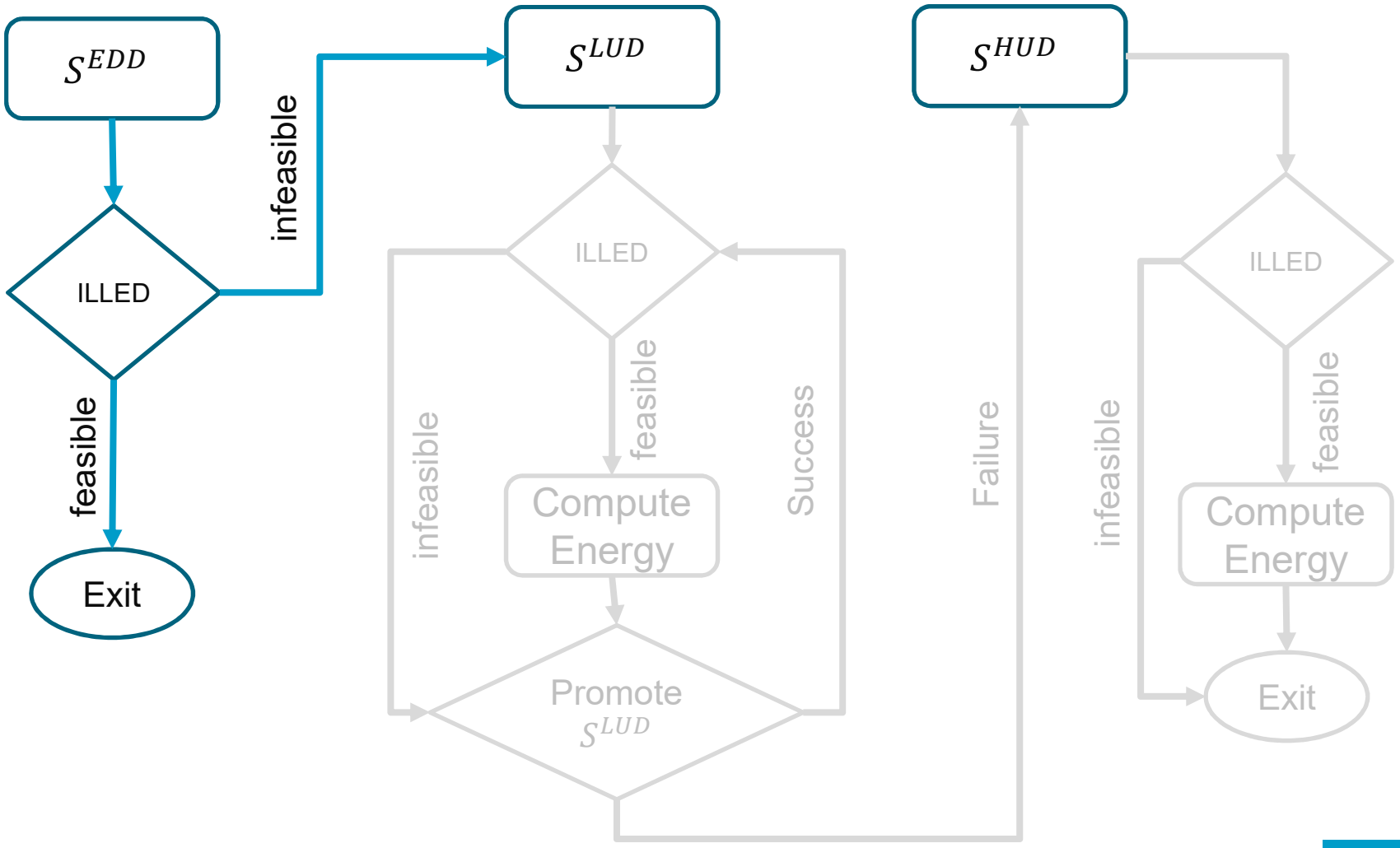
Improved least loss energy density algorithm (ILLED)

- Input: Any density difference list
  - (S^{EDD} or S^{LUD} or S^{HUD})
  - Sorted in non-increasing order



π^1 π^2 π^3 π^4

Allocation heuristics (1)

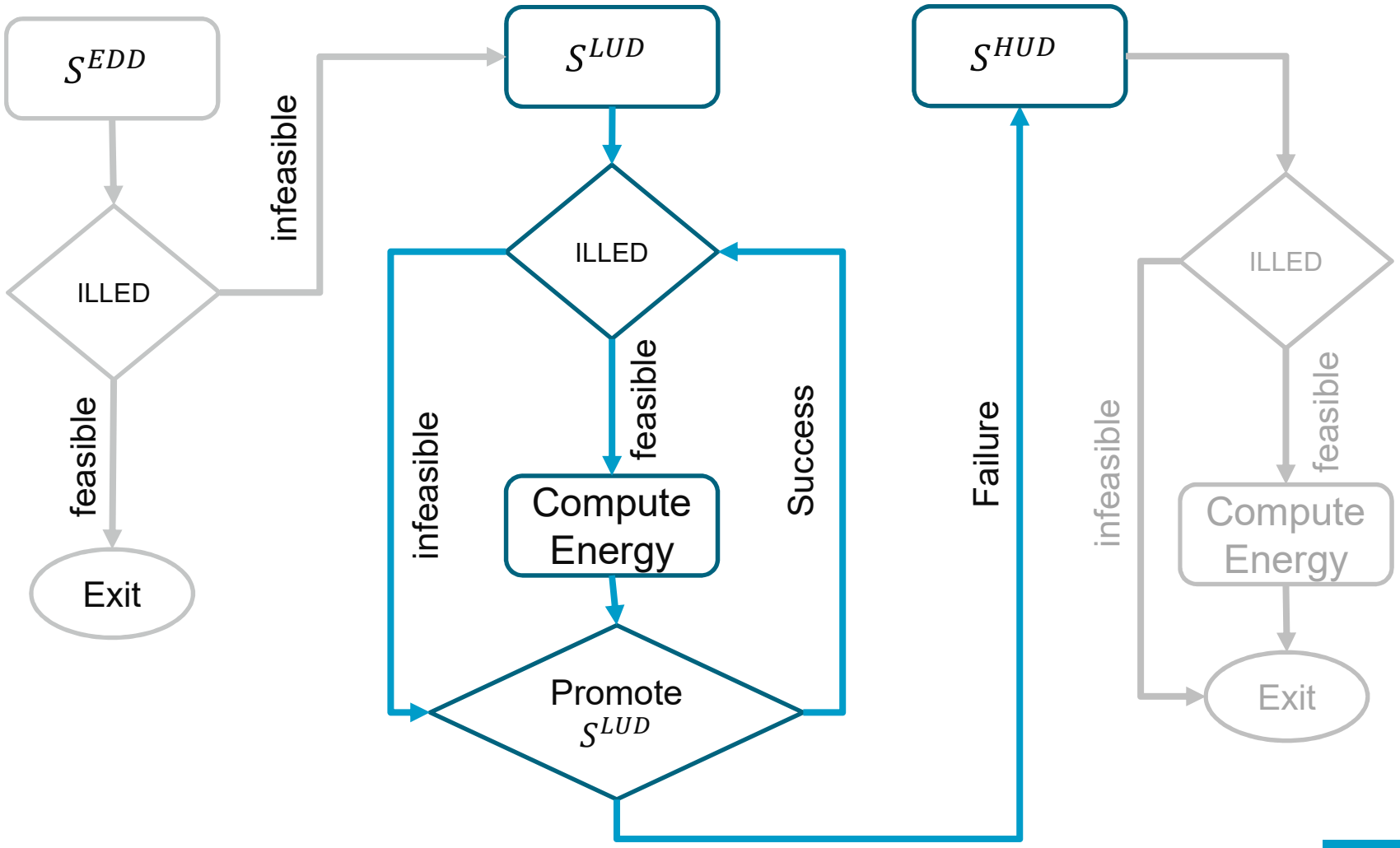


Allocation heuristics (2)

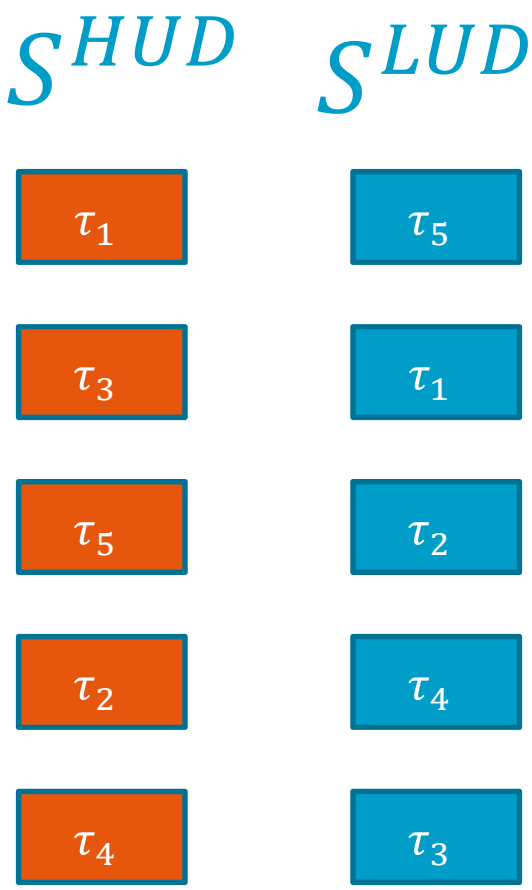
S^{EDD} S^{LUD} S^{HUD}

- Sorted list w.r.t L-mode utilisation difference
  - Improves L-mode schedulability
- Sorted list w.r.t H and L-mode utilisation difference
  - Gives preference to H-criticality tasks
  - Improves H-mode schedulability

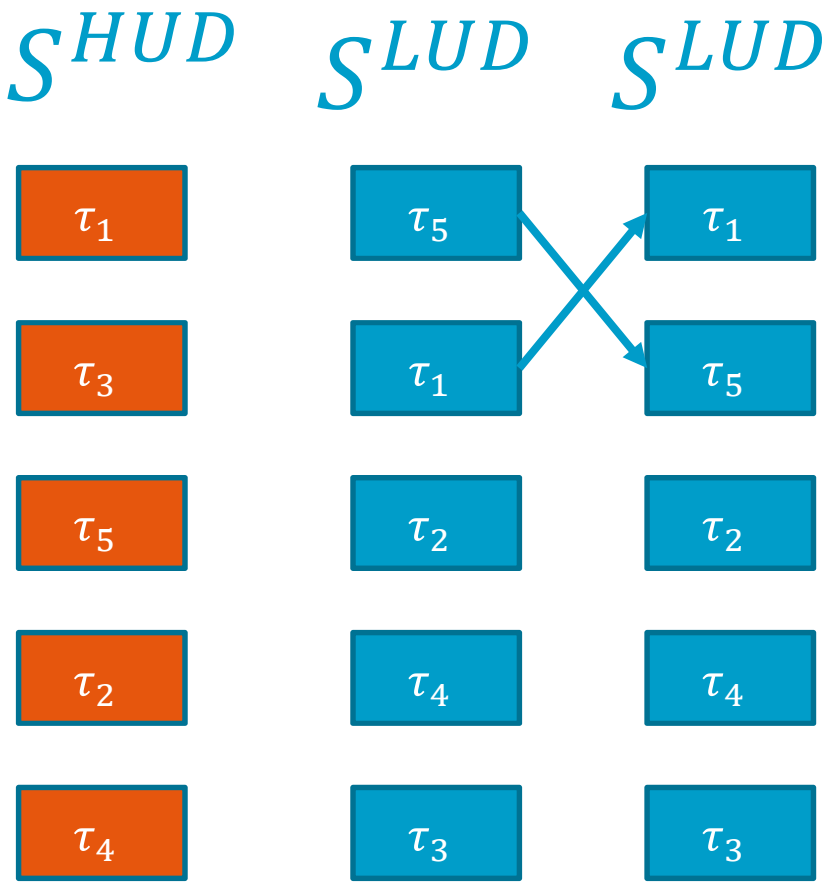
Allocation heuristics (3)



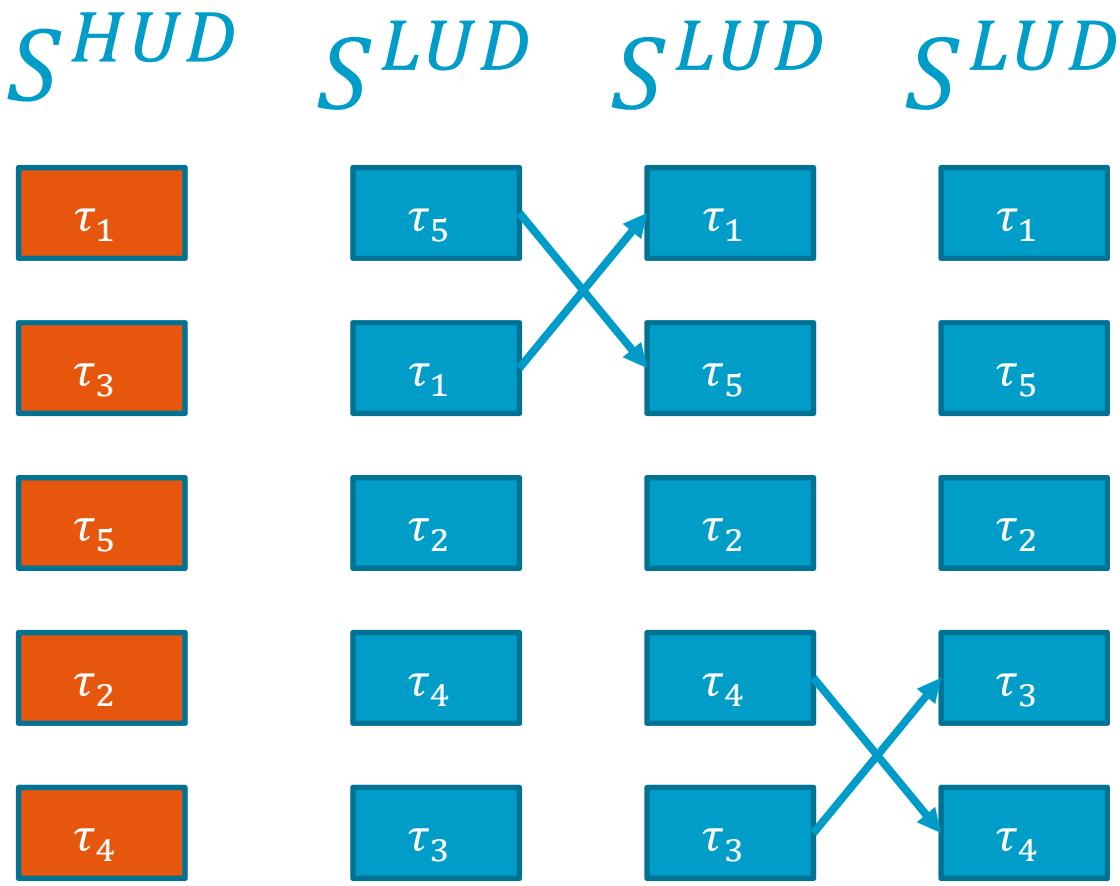
Allocation heuristics (4)



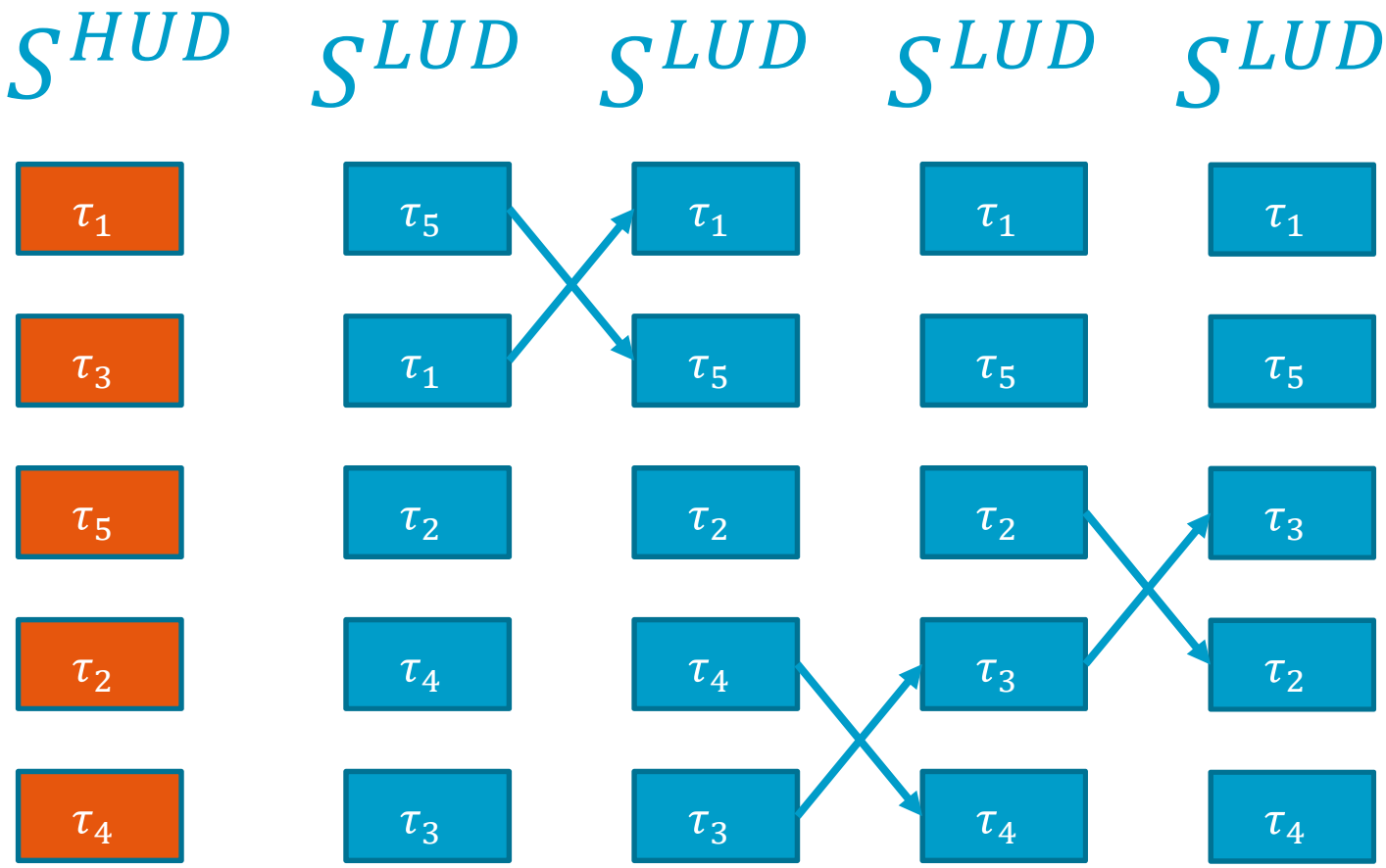
Allocation heuristics (4)



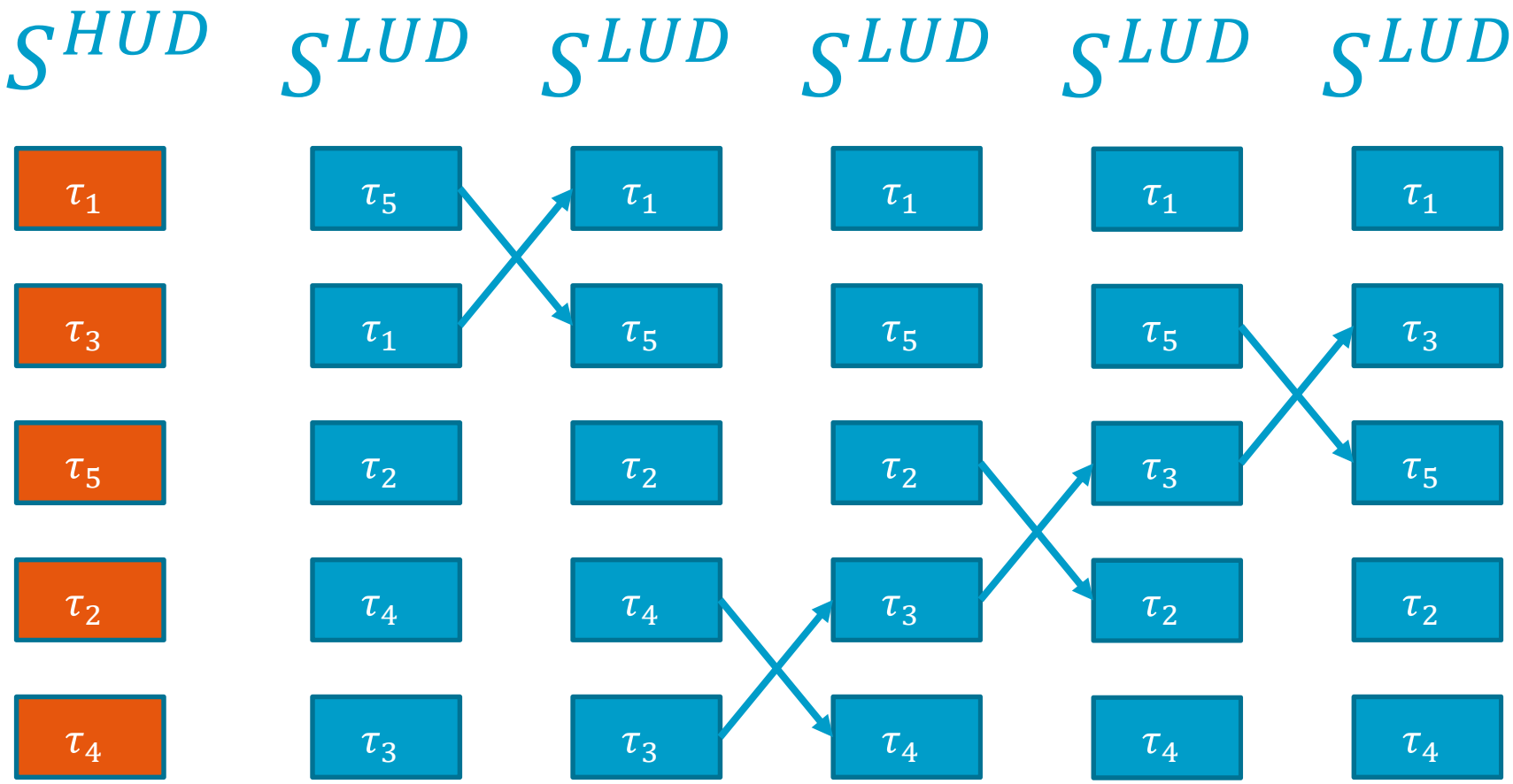
Allocation heuristics (4)



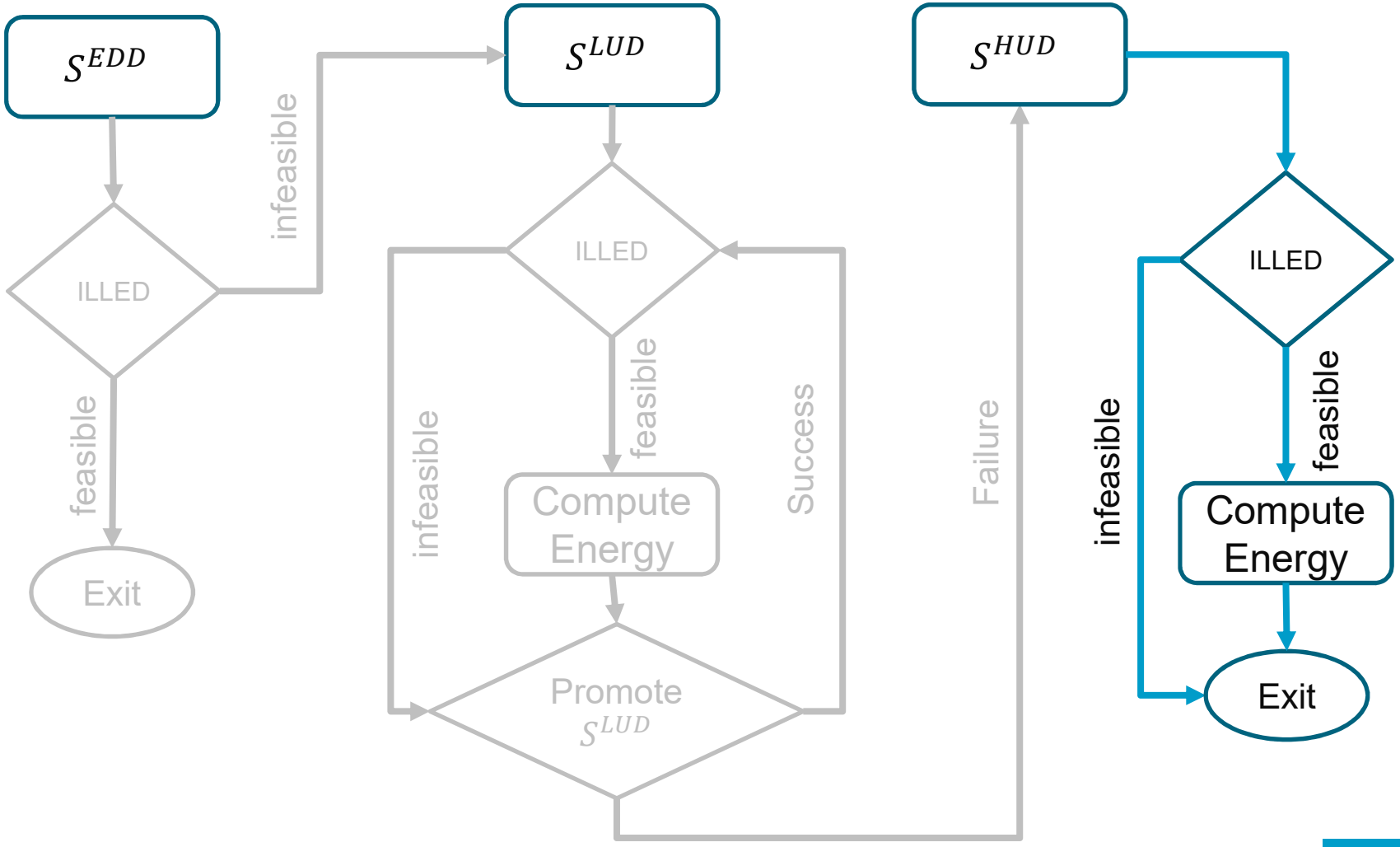
Allocation heuristics (4)



Allocation heuristics (4)



Allocation heuristics (5)



Complexity analysis

- Worst-case scenario
  - S^{EDD} is not feasible
  - All tasks are H-criticality tasks
  - S^{LUD} is in reverse order of S^{HUD}
- Iterations of ILLED = $\frac{n^2-n+4}{2}$
- ILLED has a complexity of O($n \times M$)
  - i.e., it performs $n \times M$ Ekberg and Yi analysis
- In total, $\frac{(n^3-n^2+4n)m}{2}$ Ekberg and Yi analysis

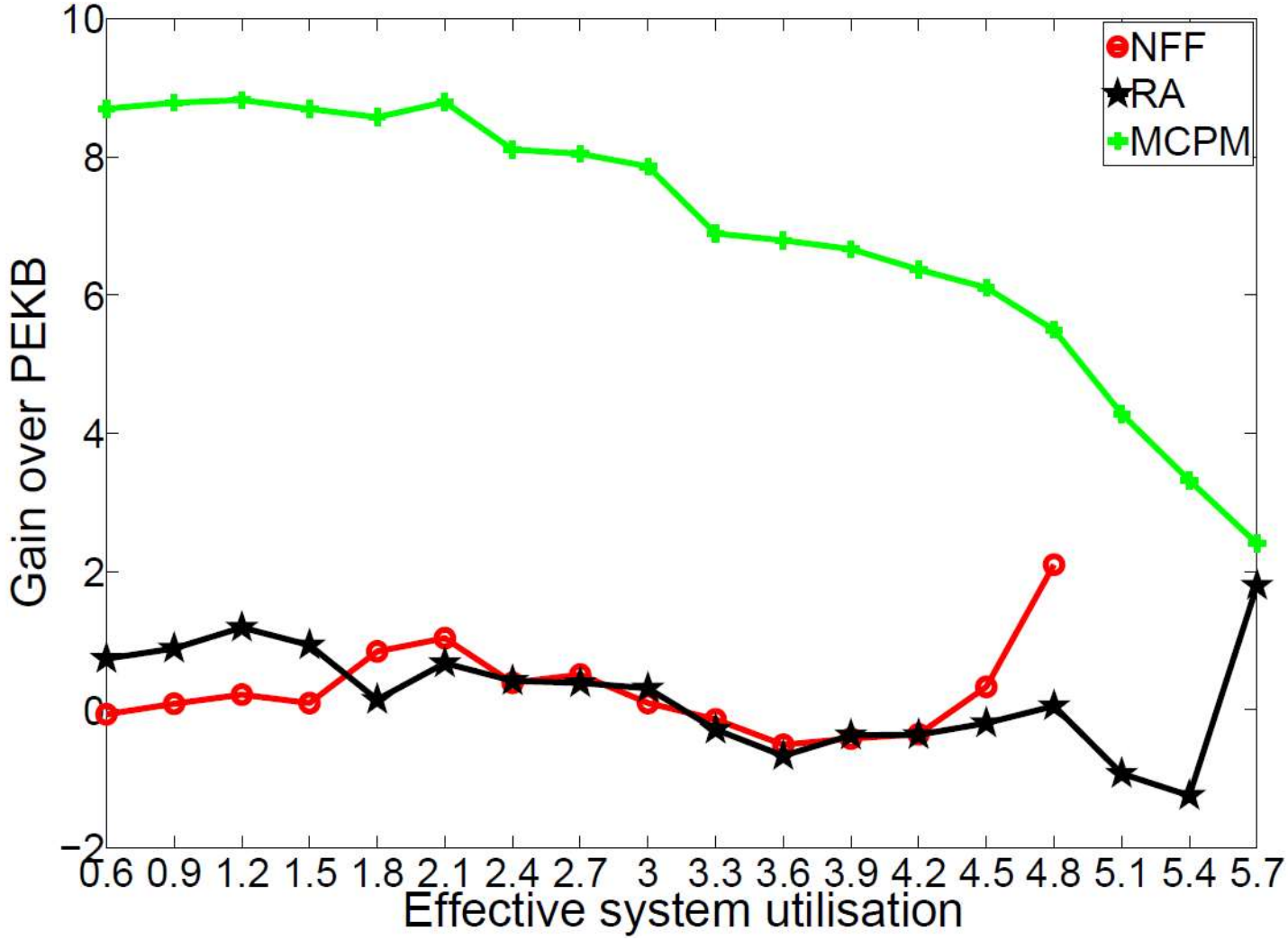
Experimental Setup

- Heterogeneous multicore platform
  - FreeScale PowerQUICC III (MPC8536)
- SPARTS simulator
  - Extended for Mixed criticality systems
  - Synthetic workload
    - Utilisation (Uunifast-discard)
    - Periods (log-uniform distribution)
    - Deadlines (Implicit deadlines)
    - H-mode WCET (transfer function)

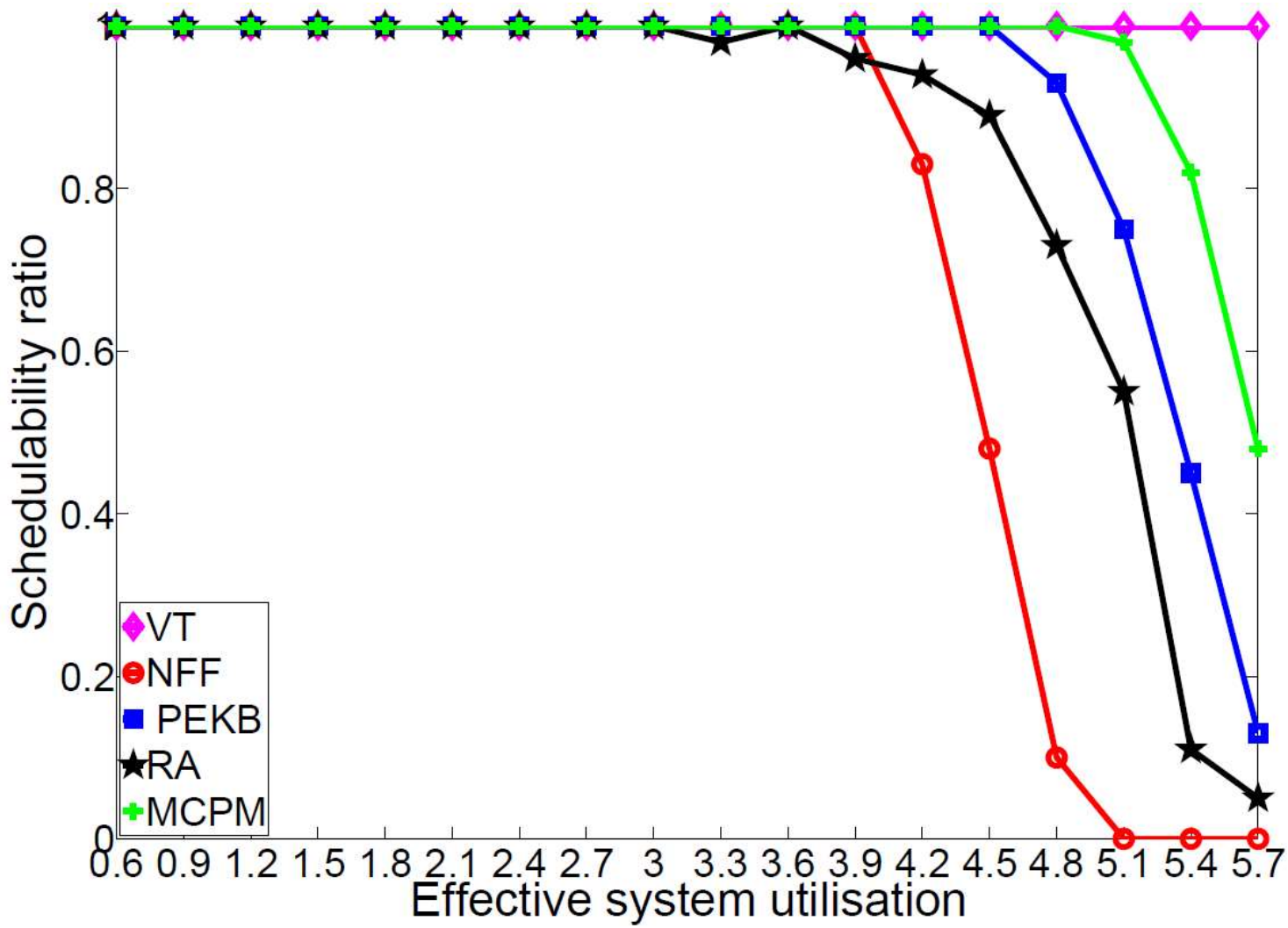
Results (1)

- Compared algorithms
  - Naïve first fit (NFF) → FF, $C_i(L)$ and $C_i(H)$
  - Partitioned Ekberg (PEKB) → FF and Ekberg and Yi analysis
  - Random allocation (RA) → random and Ekberg and Yi analysis
  - Mixed criticality power management allocation (MCPM) → our proposed heuristics

Results (2)


NFF
RA
MCPM
Gain over PEKB
10
8
6
4
2
0
−2
0.6 0.9 1.2 1.5 1.8 2.1 2.4 2.7 3 3.3 3.6 3.9 4.2 4.5 4.8 5.1 5.4 5.7
Effective system utilisation

Results (3)


Schedulability ratio
0
0.2
0.4
0.6
0.8
VT
NFF
PEKB
RA
MCPM
0.6 0.9 1.2 1.5 1.8 2.1 2.4 2.7 3 3.3 3.6 3.9 4.2 4.5 4.8 5.1 5.4 5.7
Effective system utilisation

Results (4)

Metric	H-Mode WCET multiplier	Number of cores	Number of tasks	Percentage of high criticality tasks	Variability in WCET, and energy
Energy gain	9,12%	12.01%	8.91%	8.93%	23.8%
Schedulability difference	44%	43%	44%	63%	52%
S^{EDD} $feasible$	88.35%	88.68%	88.25%	86.12%	89.52%

Conclusions and future directions

- Energy-aware task-to-core allocation
  - Heterogeneous multicore platform
  - Mixed criticality applications
- More realistic power model
- Extensive simulations show substantial gains
- Can be used for other multi objective optimisation problems
- Possible extensions:
  - Multi criticality level model
  - Integrating the effect of I/O devices

Questions and Comments